Fast Quadratic Local Meta-Models for Evolutionary Optimization of Anguilliform Swimmers

Stefan Kern, Nikolaus Hansen, Petros Koumoutsakos

Evolutionary and Deterministic Methods for Design, Optimization and Control
with Applications to Industrial and Societal Problems
EUROGEN 2007
P. Neittaanmäki, J. Périaux, and T. Tuovinen (eds.)
Jyväskylä, 11–13 June 2007

FAST QUADRATIC LOCAL META-MODELS FOR EVOLUTIONARY OPTIMIZATION OF ANGUILLIFORM SWIMMERS

Stefan Kern[⋆], **Nikolaus Hansen**[†], **and Petros Koumoutsakos**[⋆]

[⋆] Computational Science and Engineering Lab, ETH Zurich
Universitaetstrasse 6, ETH Zentrum, 8092 Zurich
e-mails: skern@inf.ethz.ch, petros@ethz.ch, web page: http://www.cse-lab.ethz.ch

[†] Projet TAO, INRIA Futurs - LRI,
Univ. Paris-Sud 91405 Orsay Cedex France
e-mail: hansen@lri.fr, web page: http://tao.lri.fr

Keywords: Optimization, Swimming, Local meta-models, Evolution Strategies

Abstract. *We combine second order local regression meta-models with the Covariance Matrix Adaptation Evolution Strategy in order to enhance it's efficiency in the optimization of computationally expensive problems. Computationally intensive direct numerical simulations of an anguilliform swimmer provide the testbed for the optimization.*

We propose two concepts to reduce the computational cost of the meta-model building. The novel versions of the local meta-model assisted Evolution Strategy are tested on benchmark problems and compared to results from literature. The results demonstrate that the use of local meta-models increases significantly the efficiency of already competitive evolution strategies and that the model building cost can be successfully reduced.

The meta-model assisted Evolution Strategy is applied to the optimization of the swimming motion of a three-dimensional, self-propelled eel-like body. The motion of the self-propelled body is determined by a set of parameters but the motion is not prescribed a-priori. Instead here we introduce the concept of identifying the swimming motion parameters from an evolutionary optimization procedure. The optimization successfully identifies a motion pattern that is 30% more efficient than an existing reference motion pattern. During the efficient swimming motion, the deformation of the body is extended along its length in a controlled fashion.

1 INTRODUCTION

The evolutionary optimization of real world problems often involves the repeated evaluation of computationally expensive fitness functions. An example of such problems involve large scale Computational Fluid Dynamics (CFD) solvers. In this work we consider the evolutionary optimization of the body motion of simulated, anguilliform (eel-like) swimmers. Starting from the pioneering work of Gray in 1933 [8], anguilliform swimming has attracted the attention of researchers from diverse scientific fields and the applications range from fundamental science to large industrial projects such as the energy harvesting eel [1].

The efficiency of Evolutionary Algorithms (EAs) for expensive problems can be improved by incorporating local or global meta-models of the fitness function [12]. Locally weighted learning methods [2] are becoming increasingly popular for building local-meta models due to their capability to adapt to the inhomogeneous distribution of the data collected by stochastic optimization algorithms [3, 14].

Local meta-models have been used to enhance the efficiency of the *Covariance Matrix Adaptation Evolution Strategy* (CMA-ES) [10] in the optimization of computationally expensive problems, resulting in the *local meta-model CMA-ES* (lmm-CMA) [14]. The lmm-CMA outperforms the standard CMA-ES on both uni- and multi-modal test functions and the meta-model does not degrade the overall performance even when the function are not modeled effectively. The investigations in [14] also revealed that *quadratic* local models are necessary in order to significantly improve the convergence speed of stochastic optimization algorithm. Lower order models are not able to provide additional information about the problem to the optimization algorithm. The computational complexity of the model building remains the main drawback of the lmm-CMA. For an optimization problem of dimension n the local *quadratic* models have $n_{\text{param}} = O(n^2)$ parameters. The exact solution of the least squares problems to determine the parameters of the local models therefore scales with $O(n^6)$ limiting the range of application to problems with $n \lesssim 15$.

We present two concepts to reduce the computational cost of building the local quadratic meta-models. On the one hand we can reduce the number of local models queries by building one local model around the mean of the offspring population to be predicted instead of building individual models for every offspring as originally proposed in [14]. On the other hand we can exploit the fact that the individual local models share mutual information, i.e. the individual least squares problems contain common data. We exploit this property of the model and propose to solve the least squares problems for the local models using the QR-factorization. The QR-factorization for a new local model can be obtained by up- and down-dating the QR-factorization of a previously computed local model containing common data. This reduces the cost of the solution of the least squares problems from $O(n^3_{\text{param}})$ to $O(n^2_{\text{param}})$ and therefore the cost for the building the local quadratic meta-model is reduced from $O(n^6)$ to $O(n^4)$.

We present variants of lmm-CMA implementing the concepts to reduce the computational cost of the meta-model building introduced above. The novel lmm-CMA variants are validated on a number of test problems and their performance is compared to the previous version. In addition, the original lmm-CMA is applied to the optimization of the swimming motion of a self-propelled eel-like body. The goal is to identify the link between swimming motion and swimming function in anguilliform (eel-like) swimming using evolutionary optimization. The computational model of the swimming simulation is based on the computationally expensive solution of the 3D Navier-Stokes equations for the incompressible viscous flow around an unsteadily deforming three dimensional eel-like body. The body motion is not prescribed a-priori but it is identified by lmm-CMA. The performance of lmm-CMA is compared to a reference optimization run using the classical CMA-ES [10].

The remainder is organized as follows: In Sect. 2 we give a brief introduction of locally weighted regression to build local meta-models in Evolution Strategies, and in Sect. 3 we revisit the lmm-CMA presented in [14]. Concepts to reduce the computational cost of the local quadratic meta-model building are presented in Sect. 4 along with lmm-CMA versions implementing these concepts. Section 5 compares the performance of the novel lmm-CMA variants on a set of benchmark problems against previous results. The application study of the meta-model assisted optimization of the swimming motion of an anguilliform swimmer is presented in Sect. 6, and the findings are summarized in Sect. 7.

2 LOCALLY WEIGHTED REGRESSION TO BUILD META-MODELS

Locally Weighted Regression (LWR) [2] attempts to fit the training data (*here*: past evaluations of the fitness function stored in a database) only in a region around the location of the query. The local models are built consecutively as queries need to be answered and therefore are intrinsically designed for growing training data sets as they occur in the course of an optimization. Thus individual models are built for every offspring to be predicted in the Evolution Strategy (ES). Given a set of points $(\boldsymbol{x}_j, y_j), j = 1, \ldots, m$, the training criterion L is minimized w.r.t. the parameters $\boldsymbol{\beta}$ of the local model $\hat{f}$ at query point $\boldsymbol{q}$ and can be written as

$$L(\boldsymbol{q}) = \sum_{j=1}^{m} \left[(\hat{f}(\boldsymbol{x}_j, \boldsymbol{\beta}) - y_j)^2 K\left(\frac{d(\boldsymbol{x}_j, \boldsymbol{q})}{h}\right)\right], \tag{1}$$

where $K(.)$ is the kernel weighting function, $d(\boldsymbol{x}_j, \boldsymbol{q})$ the distance between data point $\boldsymbol{x}_j$ and $\boldsymbol{q}$, and h is the (local) bandwidth. Investigations in [14] revealed that for a competitive EA such as CMA-ES the local polynomial meta-models $\hat{f}$ need to be of (at least) second order to be capable of speeding up the search. Consequently we choose $\hat{f}(\boldsymbol{x}, \boldsymbol{\beta}) = \tilde{\boldsymbol{x}}^T \boldsymbol{\beta}$ where $\tilde{\boldsymbol{x}}$ is a transformation of $\boldsymbol{x}$ into a basis of the polynomials of degree 2. As $\hat{f}$ is linear

in the parameters $\boldsymbol{\beta}$ we can directly weight the training points and solve

$$\boldsymbol{\beta}^\star = \operatorname*{argmin}_{\beta} \|\boldsymbol{W}\tilde{\boldsymbol{X}}\boldsymbol{\beta} - \boldsymbol{W}\boldsymbol{y}\|^2 \tag{2}$$

where $\tilde{\boldsymbol{X}} = (\tilde{\boldsymbol{x}}_\mathbf{1}, \ldots, \tilde{\boldsymbol{x}}_{\boldsymbol{m}})^T$, $\boldsymbol{y} = (y_1, \ldots, y_m)^T$, and $\boldsymbol{W} = \operatorname{diag}(\sqrt{K(d(\boldsymbol{x}_i, \boldsymbol{q})/h)})$ by using the QR-factorization of $\boldsymbol{W}\tilde{\boldsymbol{X}}$. It was show in [14] that it is advantageous to utilize the metric of the search distribution of the EA to calculate $d(\boldsymbol{x}_j, \boldsymbol{q})$ and to rescale the training data in order to assure a well conditioned least squares problem (2). For algorithms based on multivariate Gaussian mutation distributions $\mathcal{N}(\boldsymbol{m}, \boldsymbol{C})$, such as CMA-ES, the covariance matrix $\boldsymbol{C}$ naturally defines a metric that can be exploited in the calculation of d as fully weighted Euclidean distance

$$d(\boldsymbol{x}_j, \boldsymbol{q}) = \sqrt{(\boldsymbol{x}_j - \boldsymbol{q})^T \boldsymbol{C}^{-1} (\boldsymbol{x}_j - \boldsymbol{q})} = \sqrt{(\boldsymbol{x}'_j)^T \boldsymbol{x}'_j}, \tag{3}$$

$$\boldsymbol{x}'_j = \boldsymbol{C}^{-1/2}(\boldsymbol{x}_j - \boldsymbol{q}). \tag{4}$$

We choose $\tilde{\boldsymbol{x}}^T = (1, x'_1, \ldots, x'_n, x'_1 x'_2, \ldots, x'_{n-1} x'_n, {x'}^2_1, \ldots, {x'}^2_n)$ and thus the prediction reduces to $\hat{f}(\boldsymbol{q}, \boldsymbol{\beta}^\star) = \beta^\star_1$. The computation of $\boldsymbol{C}^{-1/2}$ is cheap as the eigenvalue decomposition of $\boldsymbol{C}$ is readily available in CMA-ES.

For the computation of the weights $\boldsymbol{W}$ two kernel functions K are proposed. In the original lmm-CMA [14] the bi-quadratic kernel K_2

$$K_2(\zeta) = \begin{cases} (1 - \zeta^2)^2 & \text{if} \quad \zeta < 1 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

was used. In this paper, in addition, we introduce the box kernel function K_0

$$K_0(\zeta) = \begin{cases} 1 & \text{if} \quad \zeta < 1 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

in order to facilitate up- and downdating of the QR-factorization of $\boldsymbol{W}\tilde{\boldsymbol{X}}$ (see Sect. 4).

The density of the data points collected in the course of an optimization run changes considerably, and an adaptive choice of the bandwidth h is essential. We use a *nearest neighbor bandwidth selection*, where h is set to the distance of the kth nearest neighbor data point to $\boldsymbol{q}$ and thus the volume increases and decreases in size according to the density of nearby data. In this way changes in scale of the distance function d are canceled by the choice of h, giving a scale invariant distribution of the weights to the data. For the biquadratic kernel K_2 the optimal choice for k on unimodal functions was identified in [14] to be $k = 2 \cdot (\#\text{free parameters}) = n(n+3) + 2$. In analogous test this choice for k proved also to be favourable for the box kernel K_0.

3 THE LOCAL META-MODEL CMA-ES (LMM-CMA)

The lmm-CMA [15] is a CMA-ES enhanced with meta-models based on locally quadratic regression. In meta-modeling the definition of optimal prediction needs to be consistent with the operators of the optimization algorithm [13]. Optimal prediction is usually associated with a minimum error in the quantitative approximation of the objective function by the meta-model. For rank-based EAs maintaining the fitness based ranking of the population is sufficient and therefore more appropriate. The mechanism used to incorporate meta models into CMA-ES is the *meta-model assisted ranking procedure* (MARP) originally proposed in [17]. The MARP replaces the standard evaluation and ranking procedures of the offspring population with an iterative process and results in an adaptive control mechanism determining the number of evaluated individuals in every generation: In every generation, the offspring are successively evaluated and added to the training set of the fitness function model until the (deterministic) model based selection of the parents remains unchanged in two consecutive iteration cycles. Translated to the structure of CMA-ES this implies that the predicted ranking in the μ first positions should not change for the meta-model iteration to stop. In order to be robust with large population sizes λ, often required to solve multimodal functions [9], the amount of new information added in one iteration cycle should be significant to the total information used in the model building. Therefore a batch of individuals $n_b = \max(1, \lfloor \lambda/10 \rfloor)$ proportional to λ is evaluated in every meta-model iteration. The total cost of the meta model loop can be reduced by introducing an adaptive parameter to specify the number of initial evaluations, n_{init}, performed before the model iteration loop is entered. The resulting meta-model assisted ranking procedure is outlined in Fig. 1. The computational cost of the *approximation* step in MARP is opposed to the potential reduction of expensive fitness function evaluations. In order to result in overall computational savings the cost of the repeated meta-model building must not exceed the savings obtained from the reduced number of fitness function evaluations. In [14] it has been shown that full quadratic meta-models are needed for reliable fitness predictions. Full quadratic meta-models have the disadvantage however of a computational complexity of $O(n^6)$ if the involved least squares problems are solved exactly and from scratch for every meta-model. This scaling limits the application range to moderate dimensions n. In the succeeding section we present two ideas to reduce the cost of the approximation step of the MARP.

4 REDUCING THE COMPUTATIONAL COMPLEXITY OF THE META-MODEL BUILING

The computational cost of the approximation step in the MARP can be reduced by reducing the number of models being built per saved expensive function evaluation, by reducing the cost of the solution of the least squares problems in the model building, or a combination of the two.

1 *approximate*: build $\hat{f}(\boldsymbol{x}_k), k = 1, \ldots, \lambda$ based on evaluations in training set $\mathcal{S}$
2 *rank*: based on $\hat{f}$ generate ranking$_0^\mu$ of the μ best individuals
3 *evaluate*: n_{init} best individuals based on $\hat{f}$, add to $\mathcal{S}$
4 **for** $i := 1$ **to** $(\lambda - n_{\text{init}})/n_b$ **do**
5 *approximate*: build $\hat{f}(\boldsymbol{x}_k), k = 1, \ldots, \lambda$ based on $\mathcal{S}$
6 *rank*: based on $\hat{f}$ generate ranking$_i^\mu$ of the μ best individuals
7 **if** (ranking$_{i-1}^\mu$ == ranking$_i^\mu$) **then** *(ranking of μ best remains unchanged)*
8 **break** *(exit for loop)*
9 **else** *(ranking of μ best individuals changed)*
10 *evaluate:* n_b next best unevaluated points based on $\hat{f}$, add to $\mathcal{S}$
11 **fi**
12 **od**
13 **if** $(i > 2)$ **then** $n_{\text{init}} = \min(n_{\text{init}} + n_b, \lambda - n_b)$
14 **elseif** $(i < 2)$ **then** $n_{\text{init}} = \max(n_b, n_{\text{init}} - n_b)$

Figure 1: Approximate ranking procedure that is executed in every generation to determine the fraction of points evaluated on the fitness function. The procedure is not called until sufficiently many evaluations are stored in the training set $\mathcal{S}$ to build the model; initialization of $n_{\text{init}} = \lambda$.

4.1 Reducing the number of local models built in MARP

The number of local models built per approximation step (Fig 1, line 5) can be effectively reduced by refraining from building λ models (one model for each individual to be predicted) and instead using the local models to predict more than just one point. A straight forward approach is to build only one local model around the mean of the offspring population $\langle \boldsymbol{x}_{\text{off}} \rangle$. Equation 3 changes to

$$d(\boldsymbol{x}_j) = \sqrt{(\boldsymbol{x}_j - \langle \boldsymbol{x}_{\text{off}} \rangle)^T \boldsymbol{C}^{-1} (\boldsymbol{x}_j - \langle \boldsymbol{x}_{\text{off}} \rangle)} = \sqrt{(\boldsymbol{x}'_j)^T \boldsymbol{x}'_j} \tag{7}$$

with $\boldsymbol{x}'_j = \boldsymbol{C}^{-1/2}(\boldsymbol{x}_j - \langle \boldsymbol{x}_{\text{off}} \rangle)$, and the predictions for the entire offspring population read $\hat{f}(\boldsymbol{x}_i, \boldsymbol{\beta}^\star) = \tilde{\boldsymbol{x}}_i^T \boldsymbol{\beta}^\star$, $i = 1, \ldots, \lambda$. This strategy reduces the model building cost by a factor equal to the population size. However, the scaling of the model building cost of $O(n^6)$ remains unchanged. Note that for $\lambda > k$ the local models tend to be used for extrapolation in regions where training data would be available but that were not considered in the model building due to the local bandwidth.

4.2 Up- and downdating of the QR-factorization

Neighboring local models usually share a large part of their training data. New local models can be built from existing models by adding and removing points from the training data via up- and downdating of the QR-factorization (see e.g. [19]) of the least squares problem (2) to determine the optimal model parameters. The technique used

here is based on Givens-rotations [7] to cancel out non-zero elements appearing in the QR-factorization after removal and addition of points. The cost of the updating steps scales with the dimension of the matrix squared resulting in a computational complexity with respect to the search space dimension n of $O(n^4)$. A prerequisite for the application of QR-factorization up- and downdating is a common coordinate system of all involved models and equal weights $\boldsymbol{W}$ for all points included in the regression. Equal weights are obtained by using the box kernel K_0. In order to ensure the same coordinate system for all involved models, the rescaling (4) is done offline, in contrast to the original algorithm. Consequently (3) is replaced by the Euclidian distance

$$d(\boldsymbol{x}_j, \boldsymbol{q}) = \sqrt{(\boldsymbol{x}_j - \boldsymbol{q})^T (\boldsymbol{x}_j - \boldsymbol{q})}. \tag{8}$$

The model $\hat{f}(\boldsymbol{x}, \boldsymbol{\beta}) = \tilde{\boldsymbol{x}}^T \boldsymbol{\beta}$ is built using untransformed variables, i.e. $\tilde{\boldsymbol{x}}^T = (1, x_1, \ldots$ $\ldots, x_n, x_1 x_2, \ldots, x_{n-1} x_n, x_1^2, \ldots, x_n^2)$. In order to prevent (2) from getting ill-conditioned in the course of the optimization, the training data is periodically rescaled with the current covariance matrix $\boldsymbol{C}$ of the algorithm and translated so that the mean of the latest offspring generation $\langle \boldsymbol{x}_{\text{off}} \rangle$ coincides with the origin:

$$\boldsymbol{x}_j^{(r+1)} = \left(\boldsymbol{C}^{-1/2}\right)^{(r+1)} \left(\boldsymbol{C}^{1/2}\right)^{(r)} \boldsymbol{x}_j^{(r)} - \langle \boldsymbol{x}_{\text{off}} \rangle, \tag{9}$$

where r is counter for the rescaling steps. We propose to rescale every 20 generations of the baseline optimization algorithm. The rescaling step requires that all previously built models are deleted as they become invalid for the rescaled data. The entire procedure to reduce the computational complexity of the model building in the MARP is summarized in Fig. 2.

4.3 lmm-CMA versions with reduced model building costs

We propose three novel version of lmm-CMA with reduced model building cost.

- **lmm-CMA**$_{\text{M}}$ builds only one local model to predict all individuals of an entire offspring population. The local model is centered on the mean of the offspring population and the bi-quadratic kernel K_2 is used to compute the weights $\boldsymbol{W}$.
- **lmm-CMA**$_{\text{U}}$ builds individual models for every offspring based on up- and downdating of previous QR-factorizations using the procedure summarized in Fig. 2. The up- and downdating requires the use of K_0 for the computation of the weights.
- **lmm-CMA**$_{\text{MU}}$ is a combination of both concepts where the local models are built around the mean of the offspring population by up- and downdating of QR-factorizations of previous models.

As our implementation are prototype implementations not optimized for computational efficiency, direct run time measures to determine and compare the model building costs

```
1   if mod(g, 20) == 0
2       delete all models in database, rescale training data
3   fi
4   if n_models > 0
5       compute weights using K_0
6       find model with maximum common training data in database
7       if required update steps > 5
8           goto 12
9       else
10          up- and downdate QR-factorization to solve (2)        n_up/down · O(n^4)
11      fi
12  else
13      build QR-factorization to solve (2) from scratch          O(n^6)
14  fi
15  solve (2) for β*, add to model data base                      O(n^4)
```

Figure 2: Procedure to compute the local meta models using up- and downdating of the QR-factorization pertaining to the least squares problem (2). The computational complexity of the main operations related to the solution process are indicated on the right.

of the different approaches would be meaningless. Instead, we test the performance of the different lmm-CMA versions on the Rosenbrock test-function (cf. Tab. 1) and measure the number of QR-factorizations and QR-factorization updates that the different algorithm versions perform per function evaluation saved in comparison to the standard CMA-ES. We compute statistics from 20 independent runs using the initialization regions specified in Tab. 1. The order of the model building cost is estimated by charging $O(n^6)$ floating point operations (FLOPs) for ever QR-factorization computed from scratch and $O(n^4)$ FLOPs for every QR-factorization up- or downdate. The resulting data is plotted in Figure 3. The plot shows that for $n \leq 8$ the updating technique indeed is capable of reducing scaling of the computational cost of the model building to $O(n^4)$ FLOPs per saved evaluation. For higher dimensions this advantage is lost. The reason for this effect remains unclear and is subject to future research. Switching from individual models for every offspring to a local model centered around the mean of the offspring population reduces the computational cost by about an order of magnitude which corresponds to the population size $\lambda \approx 10$ used in these dimensions.

5 PERFORMANCE COMPARISON ON TEST FUNCTIONS

The proposed novel lmm-CMA versions are investigated on a set of uni- and multimodal test-functions summarized in Table 1. The performance is assessed by averaging the number of function evaluations needed to reach $f_{\text{stop}} = 10^{-10}$ from 20 independent runs, randomly initialized in the given intervals. For the underlying CMA-ES we use

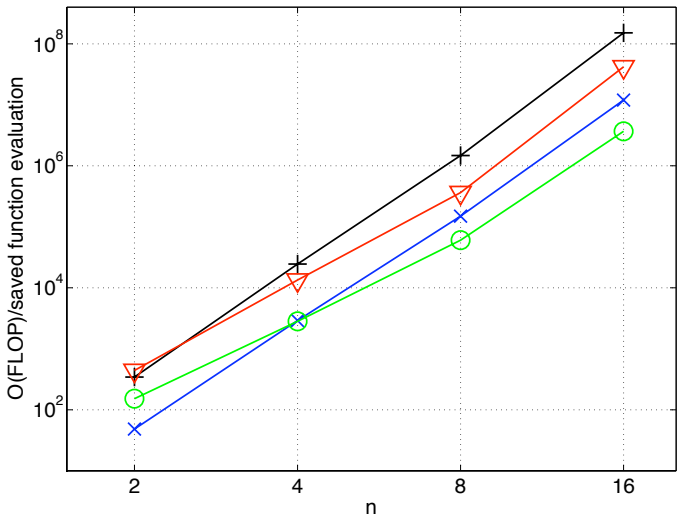

Figure 3: Order of floating point operations (FLOP) spent for the local model building per saved function evaluation (in comparison to CMA-ES) on the Rosenbrock test-function (cf. Tab. 1). The curves correspond to (+) lmm-CMA, (∇) lmm-CMA$_{\mathrm{U}}$, ($\times$) lmm-CMA$_{\mathrm{M}}$, and ($\bigcirc$) lmm-CMA$_{\mathrm{MU}}$

the standard parameter settings given in [9] except for the population size λ: for the multimodal functions we choose the optimal λ from [9, Fig. 2]. In Table 2 the results are compared to the lmm-CMA version presented in [14], the standard CMA-ES without meta-model support, the Gaussian Process Optimization Procedure (GPOP)[1] [4], and MATLAB's `fminunc`[2]. The `fminunc` implements the BFGS Quasi-Newton method with a mixed quadratic and cubic line search procedure. In the present context of black-box optimization, gradients are estimated via finite difference approximation.

On the convex quadratic f_{Sch}, all lmm-CMA versions improve CMA-ES by a factor of 5-8, and on f_{Ros} the speedup is 2-3. Compared to `fminunc`, the lmm-CMA versions at most a factor of 2.5 slower (on f_{Ros}), but on f_{Sch} they perform even better for $n \geq 8$. The performance of lmm-CMA$_{\mathrm{M}}$ and lmm-CMA on f_{Sch} and f_{Ros} are equivalent, lmm-CMA$_{\mathrm{U}}$ is 15-30% slower due to the use of the box weighting kernel K_0. The results of GPOP on f_{Sch} and f_{Ros} are competitive for $n \leq 4$. However, for larger n the Gaussian Process Regression model gets less reliable and the performance deteriorates. Note that for small n the performance gain of all lmm-CMA versions is limited by the n_b evaluations performed in every generation ($n_b = 1$ for $\lambda \leq 10$) and it generally scales well in n.

On f_{NSp} with fitness proportional Gaussian noise `fminunc` fails to converge due to the finite difference gradient estimation. In contrast, the lmm-CMA versions and CMA-ES are

[1] In GPOP the optimal size of the training data set depends on the problem and the problem dimension. For the comparison we take the best data presented in [4].

[2] We set 'LargeScale'='off', 'TolFun'=1e-10, 'TolX'=1e-15, and 'MaxFunEvals'=1e6.

Table 1: Test-functions and coordinate-wise initialization intervals.

Name	Function	Init
Noisy Sphere	$f_{\mathrm{NSp}}(\boldsymbol{x}) = (\sum_{i=1}^{n} x_i^2) \cdot (1 + \epsilon\mathcal{N}(0,1))$	$[-3,7]^n$
Schwefel	$f_{\mathrm{Sch}}(\boldsymbol{x}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$	$[-10,10]^n$
Rosenbrock	$f_{\mathrm{Ros}}(\boldsymbol{x}) = \sum_{i=1}^{n-1} \left(100 \cdot (x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right)$	$[-5,5]^n$
Ackley	$f_{\mathrm{Ack}}(\boldsymbol{x}) = 20 - 20 \cdot \exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)$ $+e - \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)\right)$	$[1,30]^n$
Rastrigin	$f_{\mathrm{Ras}}(\boldsymbol{x}) = 10n + \sum_{i=1}^{n} \left(y_i^2 - 10\cos(2\pi y_i)\right)$	$[1,5]^n$

able to cope with the noise levels as given in Table 2. The best performance is exhibited by lmm-CMA$_{\mathrm{M}}$ that has a structural advantage compared to the other versions for that particular test case. The advantage of the lmm-CMA versions compared to CMA-ES decays with increasing dimension.

On the multimodal function f_{Ack} the lmm-CMA versions perform similar and are advantageous in small dimensions ($n \leq 10$), but the improvement compared to pure CMA-ES decays with increasing n. The results on f_{Ras} reveal a reduced robustness of lmm-CMA$_{\mathrm{M}}$ on multimodal functions with large populations sizes. With the given settings k (the number of nearest neighbors to be included in the local model) becomes smaller than λ and the local-model is misused to extrapolate leading to a drastically reduced capability to find the global optimum. In contrast, lmm-CMA and lmm-CMA$_{\mathrm{U}}$ work robustly with large populations and show a slight advantage compared to CMA-ES.

6 APPLICATION: OPTIMIZATION OF ANGUILLIFORM SWIMMING

We test lmm-CMA on the optimization of simulated fish swimming and compare it's performance to the classical CMA-ES. We optimize the motion pattern of an eel-like body for swimming efficiency. Anguilliform swimming is the primary mode of locomotion for numerous aquatic species across a range of diverse taxa. Anguilliform swimmers propel themselves forward by propagating waves of curvature towards the posterior of the body and this type of locomotion is widespread among species, ranging in scales from nematodes to eels [16]. Starting from the pioneering work of Gray in 1933 [8], anguilliform swimming has attracted the attention of researchers from diverse scientific fields, ranging from neurosience to hydrodynamics [6, 20]. The striking hydrodynamic feature of anguilliform swimming is its propulsive efficiency inferred from the long range migration of the European Eel *Anguilla anguilla* without ingestion from the European cost to their spawning grounds [21].

In our ongoing work we combine control and learning of motion patterns with a fully three dimensional unsteady viscous flow simulation of the creature freely moving in the water. The time dependent motion of the body is defined by the two-dimensional de-

formation of its mid-line parameterized as traveling wave of curvature. In this paper we present the optimization of the motion with respect to swimming efficiency. Our goal is to compare the kinematics and hydrodynamics of the optimized artificial swimmer to experimental studies of real anguilliform swimmers. The simulations provide detailed information of the complete flow field enabling the quantification of the vortex formation and shedding process and enable the identification of the force distribution along the self-propelled body and their link with the kinematics of the body and the vorticity dynamics of the wake. A comprehensive survey of our work on anguilliform swimming can be found in [15].

6.1 Geometrical Model of the Eel-like Body and Parametrization of the Motion

The three dimensional geometry of the anguilliform swimmer is constructed from spatially varying ellipsoid cross sections. The length of the two half axis $w(s)$ and $h(s)$ are defined as analytical functions of the arc length s along the mid-line of the body. Following [5], the width of the body is set to 8% of its length at the head and 2% at the tail. The analytical description of $w(s)$ is divided in 3 regions:

$$w(s) = \begin{cases} \sqrt{2w_h s - s^2} & 0 \leq s < s_b \\ w_h - (w_h - w_t)\left(\frac{s-s_b}{s_t-s_b}\right)^2 & s_b \leq s < s_t \\ w_t \frac{L-s}{L-s_t} & s_t \leq s \leq L \end{cases} \tag{10}$$

where $w_h = s_b = 0.04L$, $s_t = 0.95L$, and $w_t = 0.01L$. The height $h(s)$ is set as elliptical curve with the two half axis $a = 0.51L$ and $b = 0.08L$:

$$h(s) = b\sqrt{1 - \left(\frac{s-a}{a}\right)^2} \tag{11}$$

Figure 4 illustrates the analytical description of the three dimensional geometry and the resulting three dimensional body. For simplicity the body length L is normalized to 1. This results in a surface area of the fish model of $S = 0.304$. To compute the mass m and inertial moment I_z, the body is discretized into ellipsoid disks of constant density along the mid-line s with a step size of $\Delta s = 0.001L$. The deformation of the body is treated as a composition of the solid body rotation and translation of the individual segments.

The swimming motion of the body is described by the unsteady curvature κ of the mid-line of the body parameterized as

$$\kappa(s,t) = K(s)\sin(2\pi(t/T - \tau(s))) \tag{12}$$

where $K(s)$ is the cubic spline interpolation function through the m interpolation points $K_i, i = 1, \ldots, m$, $\tau(s)$ is the phase shift along the body, and T is the cycle time. $\tau(s) =$

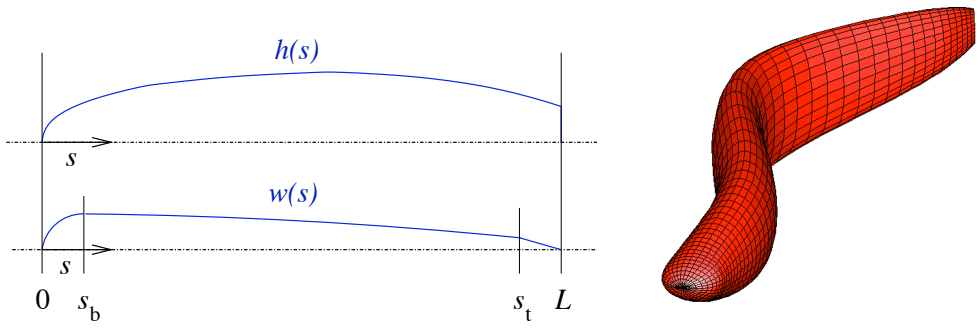

Figure 4: Illustration of the analytical description of the three dimensional body. Length of the two half axis $w(s)$ and $h(s)$ of the ellipsoid cross sections according to (10) & (11) (left), and snapshot of the resulting three dimensional deforming body (right).

$\frac{s}{L}\tau_{\text{tail}}$ is limited to be a linear function of the position s along the body. We have tried to maintain a minimum number of parameters and found that $m = 4$ interpolation points, evenly distributed along s combined with a linear phase shift suffice to allow a wide range of motion patterns.

6.2 Optimization of the Motion for Efficient Swimming

The five parameters $K_{1,\ldots,4}$, τ_{tail}, and T defining a realization of a motion pattern according to Eqn 3 and 4 are obtained through the optimization process. Defining efficiency for steady undulatory swimming is controversial [18]. The difficulty arises because the regions of drag and thrust production are distributed over the whole body and vary in the course of an undulation cycle. We relate optimizing swimming efficiency η to maximizing the amount of mean total input power $\bar{P}_{\text{total}}$ transformed into forward motion:

$$\eta = \frac{\bar{P}_{\text{forward motion}}}{\bar{P}_{\text{total}}} \tag{13}$$

The period T is constant in our optimization and we can integrate $\bar{P}_{\text{total}}$ over an entire undulation cycle to obtain the amount of work the swimmer uses per cycle W_{cycle}. The value of W_{cycle} is computed as time integral of the power output of the moving surface S of the deforming body on the surrounding fluid:

$$W_{\text{cycle}} = \int_t^{t+T} P_{\text{total}}\, dt = \int_t^{t+T} \oint_S -\bar{\bar{\boldsymbol{\sigma}}} \cdot \boldsymbol{n} \cdot \boldsymbol{u}\, dS\, dt \tag{14}$$

where $\bar{\bar{\boldsymbol{\sigma}}}$ is the viscous stress tensor, $\boldsymbol{n}$ the outer surface normal vector, and $\boldsymbol{u}$ the velocity of the moving surface. We postulate that the time integral of $\bar{P}_{\text{forward motion}}$ is related to the kinetic energy of the forward motion of the body $E_{\text{kin}} = \frac{1}{2}m\bar{U}^2$. Consequently we

assume $\eta \sim \frac{E_{\mathrm{kin}}}{W_{\mathrm{cycle}}}$, and the objective function used to optimize swimming efficiency is

$$f_\eta = \frac{m\bar{U}^2}{2W_{\mathrm{cycle}}} \tag{15}$$

We use lmm-CMA as presented in [14] to optimize f_η and compare its performance to an optimization with the standard CMA-ES [11, 10].

6.3 Equations and Numerical Method

The system of the deforming body interacting with the surrounding fluid is described by the incompressible 3D Navier-Stokes equations

$$\frac{D\boldsymbol{u}}{Dt} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\boldsymbol{u}, \quad \nabla\cdot\boldsymbol{u} = 0 \tag{16}$$

where $\frac{D}{Dt} = \frac{\delta}{\delta t} + u\cdot\nabla$ and nu denotes the kinematic viscosity of the fluid. These equations are solved along with the no-slip boundary condition on the body surface S, $\boldsymbol{u}|_S = \dot{\vec{x}}|_S$ coupled with Newtons equations of motion for the self-propelled body

$$m\ddot{\boldsymbol{x}}_c = \boldsymbol{F} \tag{17}$$

$$\dot{I}_z\dot{\phi}_c + I_z\ddot{\phi}_c = M_z \tag{18}$$

where $\boldsymbol{F}$ and M_z are the fluid force and yaw torque acting on the body surface, $\boldsymbol{x_c}$ is the position of the center of mass of the body, $\dot{\phi}_c$ its global angular velocity, m the total mass, and I_z the inertial moment about the yaw axis. The feedback of the fluid torque is limited to the yaw direction to simplify computations. The far field boundary condition is a constant static pressure modeling the fish propelling itself through an infinite tank of still fluid. The computation is carried out on a moving and deforming structured grid using a finite volume technique. The grid size is 300'000 cells for the production runs and 70'000 cells for the optimization cases. The Reynolds number $Re = \frac{\rho L\bar{U}}{\mu}$ of the simulations is in the range of $2100 - 3900$ and is comparable to the simulations of [5]. A detailed description of the computational approach including validation can be found in [15].

6.4 Optimization Performance: lmm-CMA vs. CMA-ES

We compare the performance of lmm-CMA and CMA-ES on the optimization of a a motion pattern corresponding to maximum swimming efficiency. Both optimization were started at the initial search point $K = (4.37, 2.22, 6.07, 3.07)$ and $\tau_{\mathrm{tail}} = 1.71$ obtained from preliminary investigations. The course of the optimization process for both strategies is plotted in Fig. 5; both algorithms where run with standard parameter settings as presented in [9] and [14], respectively. The lmm-CMA was stopped after 600 function evaluations. and the best parameter set found is $K = (3.45, 1.66, 6.28, 6.28)$ and $\tau_{\mathrm{tail}} = 1.73$. CMA-ES was stopped after 460 function evaluations and the best parameter set found by CMA-ES

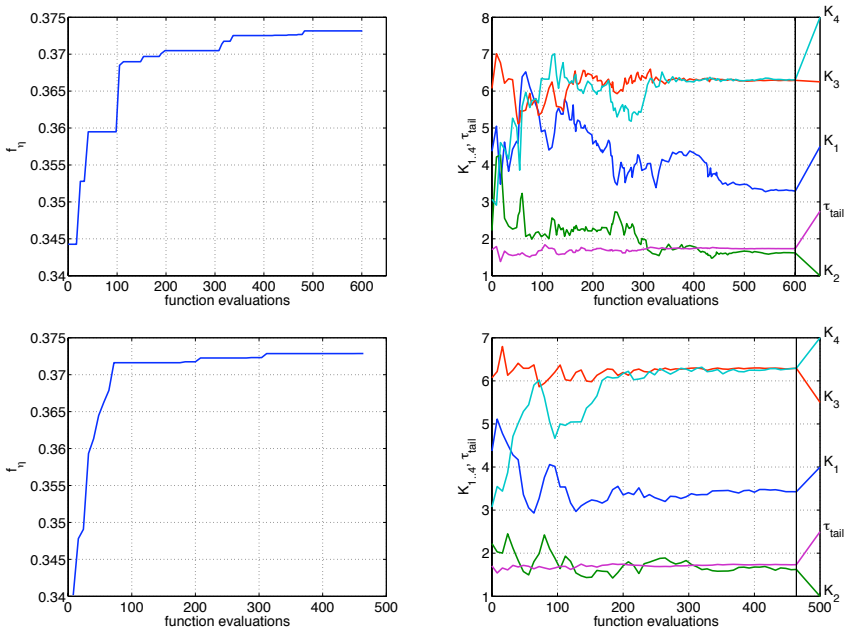

Figure 5: Convergence of the fitness value and the parameters of the efficient swimming motion for lmm-CMA (top) and CMA-ES (bottom). The strategies where manually stopped.

is $K = (3.34, 1.67, 6.28, 6.28)$ and $\tau_{\text{tail}} = 1.72$. The optimal solutions identified by the two strategies are practically identical, giving an indication of the robustness of lmm-CMA. We note that CMA-ES converged slightly faster, however the comparison of single optimization runs has only limited significance. As apparent in the tests on benchmark functions, in particular on noisy and multimodal problems, the potential speedup of lmm-CMA compared to CMA-ES can be limited. In the current application of the optimization of anguilliform swimming motions the local meta-models seem not to be able to provide accurate predictions of the fitness functions. Nevertheless lmm-CMA still works reliable without significant performance deterioration compared to CMA-ES.

6.5 Kinematics and Hydrodynamics of the Optimized Motion

We present results for the swimming motion optimized for efficient swimming and compare to a reference swimming motion described in [5]. The amplitude envelopes of the body centerline of the two motions are depicted in Fig. 6. We note that for the efficient swimming motion K_3 and K_4 converged to the maximum allowed value of 2π, indicating that swimming with higher curvature values towards the tail is beneficial for efficient swimming. At the same time this tail motion is combined with significant undulation in the anterior part of the body. The asymptotic swimming velocity of the efficient swimming is 0.33 body length per undulation cycle, while the reference swimming motion achieves 0.4 body length per undulation cycle. The input power required by the reference

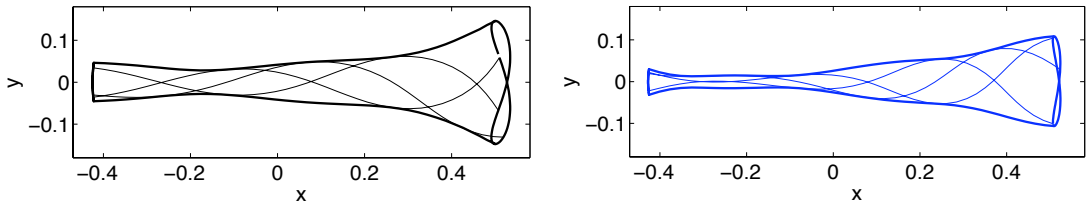

Figure 6: Amplitude envelopes of the body centerline for the reference motion pattern presented in [5] (left) and the optimized efficient swimming pattern (right).

swimming motion is 1.8 times larger than for the efficient swimming. The energy required to cover a given distance for the efficient swimming therefore is 33% smaller than for the reference swimming motion.

Flow field and wake patterns are visualized in Figs 7 and 8. The 2D cross sections of velocity and vorticity fields plotted in Fig 7 show a double row of vortex pairs forming strong lateral jets. The 3D structure of the flow consists of a double row of vortex rings with pronounced secondary structures that are visualized in Fig. 8 by plotting isosurfaces of vorticity magnitude $\|\boldsymbol{\omega}\| \equiv 2$ colored by lateral vorticity ω_y. The coloring of ω_y illustrates the boundary layer of the top and bottom part of the body being separated at the tail. In addition, signs of a longitudinal jet formed by the secondary structures can be identified. Both swimming motions show strong similarities in their wake. The smaller secondary structures in the flow of the optimized swimming motion compared to the reference motion, however, clearly indicate the increased swimming efficiency.

7 CONCLUSIONS

We have introduced two concepts to reduce the computational cost of the local metamodel building in lmm-CMA. On the one hand the number of local models queries can be reduced by building one local model around the mean of the offspring population to be predicted instead of building individual models for every offspring. On the other hand up- and downdating of the QR-factorizations pertaining to the least squares problems to be solved for the regression model can be used to exploit common data. We have presented novel versions of lmm-CMA implementing these concepts. We assessed the reduction in the model building cost as well as the performance in comparison with the original version of lmm-CMA.

Building only a single local model around the mean of the offspring population reduces the model building cost proportional to the populations size while the scaling in dimension n remains $O(n^6)$. Unfortunately this approach drastically reduces the robustness of lmm-CMA on highly multi-modal problems requiring large populations. The success probability to find the global optimum vanishes for lmm-CMA$_{\mathrm{M}}$ on the Rastrigin test-function for $n > 2$. On the other test functions lmm-CMA$_{\mathrm{M}}$ performs comparable to the original lmm-CMA with a slight advantage on the Noisy Sphere problem.

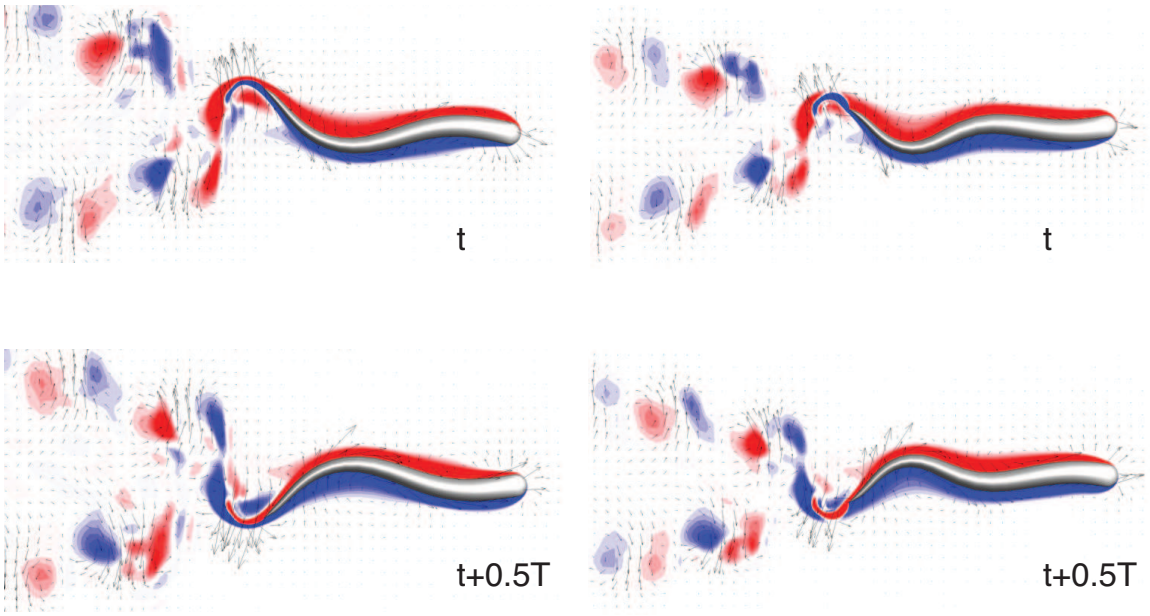

Figure 7: Velocity field and vorticity normal to the image plane ω_z for the reference swimming motion (left) and the efficient swimming motion (right).

lmm-CMA$_\mathrm{U}$ is able to reduce the scaling of the computational complexity of the model building to $O(n^4)$ for $n \leq 8$. For larger n it again deteriorates to $O(n^6)$. The reason for this effect remains unclear and is subject to future research. The speedup potential of lmm-CMA$_\mathrm{U}$ is slightly reduced compared to the original lmm-CMA due to the non-smooth kernel function K_0 used in the computation of the weights.

We have applied lmm-CMA to the optimization of 3D simulations of optimized self-propelled anguilliform swimming in a viscous incompressible fluid. An inverse design procedure is employed in order to identify an efficient swimming pattern: the swimming motions are not described a-priori but are obtained through an evolutionary optimization. This enables a systematic exploration of the motion parameter space and the identification of an efficient swimming motion involving a harmonious undulation of the whole body. The swimming kinematics and wake structures obtained in the present simulations agree well with experimental data and further elucidate the mechanisms of anguilliform swimming. On this particular application we could not identify a speedup of the convergence of lmm-CMA compared to CMA-ES.

Financial support of the Swiss National Science Foundation is acknowledged.

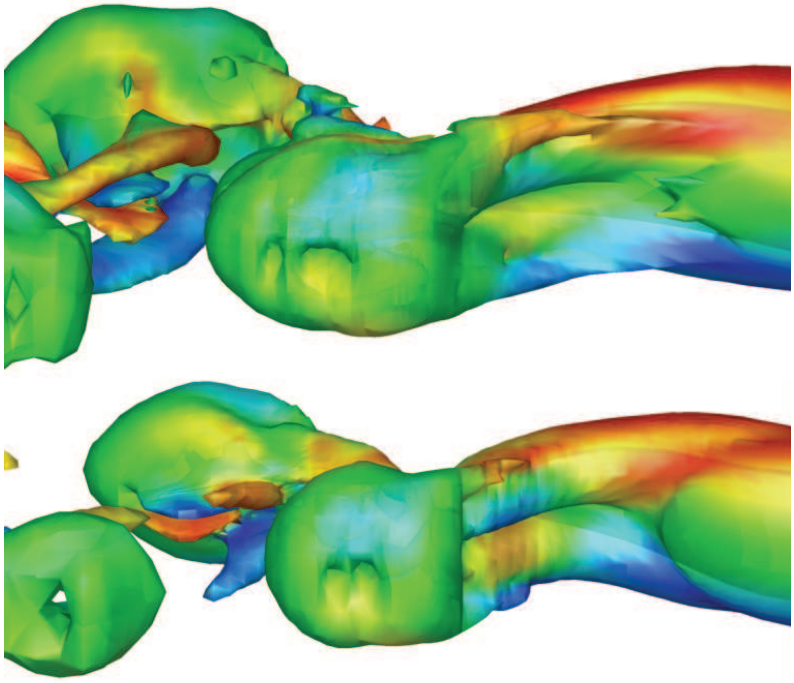

Figure 8: Flow structure at the tail visualized by isosurfaces of vorticity magnitude $\|\boldsymbol{\omega}\| \equiv 2$ colored by contours of vorticity in lateral direction ω_y for reference motion pattern (top) and the efficient swimming motion (bottom).

REFERENCES

[1] J. J. Allen and A. J. Smits. Energy harvesting eel. *Journal of Fluids and Structures*, 15(3):629–640, April 2001.

[2] C. G. Atkeson, A. W. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11(1):11–73, 1997.

[3] J. Branke and C. Schmidt. Faster convergence by means of fitness estimation. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 9(1):13–20, 2005.

[4] D. Büche, N. N. Schraudolph, and P. Koumoutsakos. Accelerating evolutionary algorithms with gaussian process fitness function models. *Systems, Man and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 35(2):183–194, 2005.

[5] J. Carling, T. Williams, and G. Bowtell. Self-propelled anguilliform swimming: simultaneous solution of the two-dimensional navier-stokes equations and newton's laws of motion. *Journal of Experimental Biology*, 201(23):3143–3166, 1998.

[6] O. Ekeberg. A combined neuronal and mechanical model of fish swimming. *Biological Cybernetics*, 69(5):363–374, 1993.

[7] W. Givens. Computation of plane unitary rotations transforming a general matrix to triangular form. *Journal of the Society for Industrial and Applied Mathematics*, 6(1):26–50, 1958.

[8] J. Gray. Studies in animal locomotion: I. the movement of fish with special reference to the eel. *Journal of Experimental Biology*, 10(1):88–104, 1933.

[9] N. Hansen and S. Kern. *Evaluating the CMA Evolution Strategy on Multimodal Test Functions*. 2004.

[10] N. Hansen, S. D. Muller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary Computation*, 11(1):1–18, 2003.

[11] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[12] Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 9(1):3–12, 2005.

[13] Y. Jin, M. Hüsken, and B. Sendhoff. Quality measures for approximate models in evolutionary computation. In A. M. Barry, editor, *GECCO 2003: Proceedings of the Bird of a Feather Workshop, Genetic and Evolutionary Computation Conference*, pages 170–173. AAAI, 2003.

[14] S. Kern, N. Hansen, and P. Koumoutsakos. Local meta-models for optimization using evolution strategies. In T. Runarsson, H.-G. Beyer, E. Burke, J. Merelo-Guervoós, L. Whitley, and X. Yao, editors, *Parallel Problem Solving from Nature - PPSN IX*, pages 939–948. Springer, 2006.

[15] S. Kern and P. Koumoutsakos. Simulations of optimized anguilliform swimming. *J Exp Biol*, 209(24):4841–4857, 2006.

[16] U. K. Muller, J. Smit, E. J. Stamhuis, and J. J. Videler. How the body contributes to the wake in undulatory fish swimming: Flow fields of a swimming eel (anguilla anguilla). *JOURNAL OF EXPERIMENTAL BIOLOGY*, 204(16):2751–2762, Aug 2001.

[17] T. P. Runarsson. Constrained evolutionary optimization by approximate ranking and surrogate models. *Parallel Problem Solving from Nature - PPSN VIII*, pages 401–410, 2004.

[18] W. W. Schultz and P. W. Webb. Power requirements of swimming: Do new methods resolve old questions? *Integrative and Comparative Biology*, 42(5):1018–1025, 2002.

[19] A. Sjo. Updating techniques in recursive least-squares estimation. Technical report, Lund University, 1992.

[20] E. D. Tytell and G. V. Lauder. The hydrodynamics of eel swimming: I. wake structure. *Journal of Experimental Biology*, 207(11):1825–1841, 2004.

[21] V. van Ginneken, E. Antonissen, U. K. Muller, R. Booms, E. Eding, J. Verreth, and G. van den Thillart. Eel migration to the sargasso: remarkably high swimming efficiency and low energy costs. *Journal of Experimental Biology*, 208(7):1329–1335, Apr 2005.

Table 2: Average number of function evaluations and standard deviations to reach f_{stop} of lmm-CMA and it's versions lmm-CMA$_{\text{M}}$ and lmm-CMA$_{\text{U}}$ versus CMA-ES, GPOP [4], and `fminunc`. For the multimodal functions, the numbers are divided by the probability to find the global optimum given in brackets. `fminunc` diverges on f_{NSp} (†) and has a vanishing probability to converge to the global optimum on f_{Ack} and f_{Ras} for the given initialization region. The noise levels for f_{NSp} are $\epsilon = 0.35$, 0.35, 0.18, and 0.13 for $n = 2$, 4, 8, and 16, respectively.

Func	n	λ	lmm-CMA			lmm-CMA$_{\text{M}}$			lmm-CMA$_{\text{U}}$			CMA-ES			GPOP	`fminunc`		
$f_{\text{Sch}}(\boldsymbol{x})$	2	6	81	±5		79	±5		128	±18		391	±42		40	**24**	±5	
	4	8	145	±7		136	±6		173	±21		861	±53		110	**96**	±7	
	8	10	**282**	±11		299	±13		327	±20		2035	±93		440	428	±22	
	16	12	**626**	±17		722	±16		769	±27		5263	±115		6000	1684	±37	
$f_{\text{Ros}}(\boldsymbol{x})$	2	6	263	±87	(1.0)	235	±69	(1.0)	386	±140	(1.0)	799	±119	(1.0)	180	**119**	±38	(1.0)
	4	8	674	±103	(1.0)	584	±134	(.90)	983	±135	(1.0)	1973	±291	(.95)	700	**344**	±52	(.85)
	8	10	2494	±511	(.90)	1829	±240	(.95)	2691	±385	(.95)	6329	±747	(.85)	2500	**1057**	±119	(.95)
	16	12	7299	±1154	(1.0)	7241	±487	(.90)	8380	±742	(1.0)	16388	±1414	(.95)	14000	**3628**	±226	(.90)
$f_{\text{NSp}}(\boldsymbol{x})$	2	6	184	±24		**138**	±20		199	±26		372	±39		-	†		
	4	8	503	±56		**326**	±39		521	±47		855	±93		-	†		
	8	10	1179	±103		**797**	±52		1200	±69		1645	±84		-	†		
	16	12	2700	±112		**2412**	±82		3024	±98		3073	±94		-	†		
$f_{\text{Ack}}(\boldsymbol{x})$	2	5	308	±33	(.95)	**301**	±29	(.90)	324	±30	(1.0)	728	±51	(.95)	-	∞	(0.0)	
	5	7	1095	±81	(1.0)	**1005**	±73	(.85)	1031	±74	(1.0)	1767	±74	(1.0)	-	∞	(0.0)	
	10	10	3029	±106	(1.0)	**2758**	±402	(1.0)	2981	±80	(1.0)	3637	±110	(1.0)	-	∞	(0.0)	
	20	10	8150	±196	(1.0)	10182	±1445	(.80)	8296	±106	(1.0)	**6155**	±409	(1.0)	-	∞	(0.0)	
$f_{\text{Ras}}(\boldsymbol{x})$	2	50	**1360**	±264	(.85)	12715	±4592	(.05)	1385	±279	(.80)	1982	±325	(.85)	-	∞	(0.0)	
	5	140	**7320**	±1205	(.85)	∞	(0.0)		8992	±1309	(0.65)	8486	±1160	(.85)	-	∞	(0.0)	
	10	500	29250	±2769	(1.0)	∞	(0.0)		**26972**	±2625	(1.0)	40152	±5409	(.95)	-	∞	(0.0)	